OET004 – MICROPROCESSORS/MICROCONTROLLERS TAG RUBRIC **November 2019**

Credit Hours: 3-4 Semester Hours

Prerequisite: Digital Electronics (OET002)

Related TAG: Electrical Engineering Technology

General Course Description: This course includes microprocessor/microcontroller architecture, instruction sets, software development, interrupt handling, memory, interfacing techniques, and hardware used in control applications designed with microprocessor/ microcontrollers. Includes hands-on labs.

Student learning outcomes marked with an asterisk (*) are essential and must be met.

TAG Learning Outcomes	Applied skills strongly demonstrated	Some applied skills present	Little applied skills present	No applied skills present
1. Explain microprocessor architecture.*	• Describe the steps in executing an instruction and how each part of the architecture interreacts for the key parts and programmer accessible CPU registers.		• Explain the function of the key parts. • Explain the purpose of each programmer accessible CPU register.	• Describe the key parts of a Microprocessor architecture- CPU, memory, three busses and ports. • State the programmer accessible CPU registers.
2. Utilize assembly language programming to develop code for a microprocessor.*	• Design, code and test an assembly language program including iteration • Design, code and test an assembly language program that interacts with a devise connected to a port. • Design, code and test an assembly language, program using assembler directives.	• Design, code and test a simple straight-line assembly language program. • Design, code and test an assembly language program including iteration. • Describe assembler directives and what they are used for. • Write an assembly language program using	• Translate machine instructions to assembler instructions and vice versa. • Assemble, download and run an existing program. • Explain the function of the stack and subroutines. • State several ways to create needed timing	• Explain what assembly language is and how it relates to machine language. • Explain what an assembler is. • Define stack and subroutines and why they are important. • Explain why time consideration is important in

	• Design, code and test an assembly language program using the stack and subroutines. • Design, code and test an assembly language program using time delays. • Download and debug an assembly language using interrupts.	the stack and subroutines • Write an assembly language program using programmer accessible CPU registers. • Write an assembly language program using time delays. • Write an assembly language program using interrupts.	and what each might be used for. • Explain why interrupts are important and what they are used for.	Microcontroller programming. • Describe what interrupts are.
3. Explain and utilize bus timing diagrams.*	• Sketch a timing diagram for several assembly language instructions including the individual control signals.	• Identify and give the function for the individual control signals in the control bus.		
4. Demonstrate an understanding of and applications for bus structures.*	• Describe how each bus is used in executing a write-to-port instruction.	• Describe how each bus is used in an instruction fetch and execute.	• Describe the function of each bus.	• List the three main busses.
5. Utilize memory technologies and interfacing in microprocessors.*	• Describe the sequence for writing and reading from a memory chip in terms of the memory connections.	• Describe the connections to a memory chip and what each does.	• Describe how each memory technology works and what each is typically used for.	• Identify the various memory technologies
6. Implement input/output (I/O) systems, I/O interface requirements, and interrupt based I/O.*	• Test the program using lights and switches. • Debug using debugging tools in the monitor or operating system. • Download and test a program using interrupt based I/O.	• Assemble and debug at the assembler level a program that interfaces with lights and switches. • Write, assemble and debug at the assembler level a program using interrupt based I/O	• Code a program that interfaces with lights and switches. • Design a program using interrupt based I/O.	• Design a program that interfaces with lights and switches. • Explain how interrupt based I/O works.

7. Utilize direct memory access (DMA) in microprocessor applications.*	• Design, build and test a system using DMA and DMA controller.	• Describe the function of a DMA controller.	• Identify the different approaches to DMA.	• Describe the purpose of DMA.
8. Utilize microprocessors/microcontrollers in a variety of applications.*	• Test and debug a system using two or more peripherals.	• Assemble and debug the code at the assembler level using two or more peripherals.	• Code the program using two or more peripherals and assemble the electronic hardware.	• Design a system using two or more peripherals at the block diagram and flow chart level
Note that this TAG should be updated to include programming with the C language since most of the programming now is done with C or C-like languages.				